



Contribution ID: 17

Type: Poster

Design and Implementation of Custom DMA Controller for the ALICE CRU, to Optimize Data Transfer Reducing the CPU Utilization

Wednesday 13 September 2017 17:45 (15 minutes)

The CRU (Common Readout Unit) is the new readout card that will be used in ALICE during Run 3. The card will receive detector data and it will store the information in the memory of the PC through DMA. To handle the high data throughput an Altera Arria 10 FPGA has been installed on the CRU. A custom DMA controller has been developed to optimize the DMA data transfer reducing the CPU utilization. The paper describes the details of implementation and the communication between software and firmware. It also shows the results obtained during the test concerning data throughput and PCIe usage

Summary

ALICE (A Large Ion Collider Experiment) is preparing a major upgrade and starting from 2021, it will collect data with several upgraded sub-detectors. The ALICE read out system will be upgraded as well, with a new detector data link called GBT (Giga Bit Transceiver) and new PCIe (Peripheral Component Interconnect) gen.3 x16, interface card called CRU. The card will receive several GBT links in input, up to a maximum of 36 and it will store the data in the memory of the PC through two PCIe gen.3 x8 performing DMA (Direct Memory Access). The raw data bandwidth of PCIe in gen3.x16 mode is 128 Gb/s. The CRU is equipped with an Altera Arria 10 FPGA that provides two PCIe endpoints gen.3 x8 to accommodate the high incoming data throughput. DMA is used to transfer data to the main memory of the server hosting the CRU. In Run 3, the readout servers will use the CRU to collect data from the detectors and store it in the memory.

The DMA engine works based on descriptor that contains two types of information, page size and location of the transfer. The page size of each descriptor supported by the DMA engine is of the order of Kbytes. One single DMA transaction belongs to one single descriptor. The DMA controller pulls descriptor one after another from host memory and push them towards the DMA engine to produce pipelined DMA transfers in order to achieve high performance (85% of the raw data bandwidth). The controller also updates the host side status memory associated with each descriptor when single descriptor gets executed i.e. successful transfer of one page. Software checks the status memory continuously and makes another page available for next DMA transaction upon arrival of status for the previous one. So, in principle, the status memory is updated after every page transfer and software has to react at that pace to maintain the pipeline. This high interaction rate due to small page size leads to high CPU utilization. The ALICE readout software running on the server that hosts the CRU can't be dedicated entirely to the DMA transfer as there are many other tasks that must run in parallel such as data online processing and data transport over the network. For this reason, the new readout software will allocate multiple big buffers (size ~ few Mbytes to Gbytes), called super pages and share the source addresses and availability of each one with the CRU. Based on that the custom DMA controller will generate descriptors page wise and fill those super pages as per the availability. The firmware provides the number of successful DMA transfers done to the software who decides which super page can be re-allocated and used again by the CRU. In this way, the CPU utilization is reduced to the minimum without reducing the DMA performance. This approach will also reduce the logic and memory used for managing descriptor and status memory, reducing the PCIe transaction rate considerably.

Primary author: MUKHERJEE, Sanjoy (Bose Institute (IN))

Co-authors: COSTA, Filippo (CERN); BOESCHOTEN, Pascal (Ministere des affaires etrangeres et europeennes (FR)); IMREK, Jozsef (Hungarian Academy of Sciences (HU))

Presenter: IMREK, Jozsef (Hungarian Academy of Sciences (HU))

Session Classification: POSTER Session

Track Classification: Programmable Logic, Design Tools and Methods